موضع اليوم حول:



الجزء الثاني



بسم الله الرحمن الرحيم

و الصلاة و السلام على أشرف المرسلين سيدنا محمد خاتم الأنبياء و المرسلين

أما بعد

س: ما هي "سين جيم"؟

ج: "سين جيم" هي سلسلة من المواضيع الخفيفة و الهامة لكل مبرمج. إذ أننا في كل مرة سنتناول فكرة معينة، مبدأ معين، تعريف معين، و يكون من الأمور التي تصادفنا كمبرمجين مع محاولة التركيز على الشرح و مبدأ العمل و خطوات الاستعمال بالترتيب في حين سنحاول قدر الإمكان استعمال الخطوات بدل الأكواد لتفادي التعارض بين اللغات. أما في حالة كانت هناك ضرورة تستدعي استخدام الأكواد البرمجية للغة معينة فسنضعها للإيضاح فقط.

أتمنى أن تنال السلسلة رضاكم و أرجو أن لا تبخلوا علي بأي سؤال يعتريكم في أمر من أمور البرمجة كي أضعه في موضوع بإذن الله ليستفيد الجميع. و أجري و أجركم على الله و الله من وراء المقصد.

أخوكم سمير علوي

س: ما هو موضوع اليوم؟

ج: موضوع اليوم هو تكملة للموضوع الذي بدأناه سابقا و هو موضوع XML. فبعد أن تحدثنا عن الشكل العام له. سنتحدث اليوم عن القيم و كيف تمثل و كيف تحفظ بين الوسوم أو في متغيرات.



س: بداية. ماهي أنواع القيم التي يمكن للـ XML تخزينها.

ج: كما قلنا سابقا فإن XML بنية خاملة و ليس كائنا نشيطا فهو يحفظ القيم فقط و القيم التي يحفظها تكون من الأنواع البسيطة و هي إما قيم رقمية و إما قيم حرفية. و لا يمكنه أن يحفظ قيما بشكل آخر مثل الألوان أو الصور أو الأصوات..إلخ



س: و لكنك قلت في المرة السابقة أنه يمكن تخزين أي شيء في ملف xml فكيف يمكن ذلك إن لم يكن باستطاعته حفظ القيم المركبة؟

ج: الأمر بسيط جدا. هذه الأشياء يتم تمثيل قيمها المركبة بقيم بسيطة حرفية أو رقمية. و سواء يتم ذلك آليا أو يدويا



س: كيف ذلك؟

ج: حسب الحالات فهناك تحويلات آلية و أخرى يدوية تستدعي تدخل المبرمج و سأوضح ببعض الأمثلة البسيطة

- **النوع المنطقي "Boolean"** يتم تمثيل القيمة "صح" بالكلمة true و القيمة "خطأ" بالكلمة false فالتمثيل هنا يكون آليا سواء عند الكتابة في ملف xml أو القراءة منه

- **التعدادات**: في العادة التعدادات هي بنية بها مجموعة من الثوابت التي تكون قيمها أعدادا تامة أي أنها لا تحتوي أرقاما عشرية. فالمتغيرات من هذا النوع يتم حفظ اسمها بدل قيمتها في ملف xml. فنأخذ مثالا بتعريف تعداد جديد ثم نعرف متغير من ذلك التعداد و نعطيه قيمة.

```
Enum DayNames As Byte
    Saturday = 0
    Sunday = 1
    Monday = 2
    Tusday = 3
    Wednesday = 4
    Thursday = 5
    Friday = 6
End Enum
Dim x As DayNames = DayNames.Monday
```

و النتيجة تكون كالآتي:

```
<DayNames>Monday</DayNames>
```

يمكننا ملاحظة أن النوع المنطقي في الحقيقة ما هو إلا تعداد حيث القيمة 0 تعني خطأ و أي قيمة تختلف عن الصفر تعني صح

- **التاريخ**: التاريخ نوع مركب من حقول هي: السنوات، الأشهر، الأيام، الساعات، الدقائق، الثواني، الأجزاء من الألف من الثانية و هي كلها أعداد تامة موجبة و هي بذلك قابلة للتحويل الآلي غير أن xml لا يحفظها كحقول بل بشكل جملة يبدأ فيها بالسنوات ثم الأشهر ثم الأيام بينها علامة "-" (ناقص). ثم يأتي حرف "T" بمعنى Time. و يكتب بعدها الساعات ثم الدقائق ثم الثواني بينها الرمز ":"، ثم الأجزاء من الألف من الثانية و يليها الفارق الزمني عن خط غرينيتش. فلو أردت تمثيل الوقت الحالي باستعمال الدالة Now فالنتيجة xml ستكون بالشكل:

```
<dateTime>2011-09-19T14:24:28.2093437+01:00</dateTime>
```

- **الهياكل**: و هي نوع مركب من عدة حقول يشبه الكلاس في كثير من الأمور و هو ما نسميه Enregistrement حيث يتوجب علينا تعريف قيم الحقول للمتغير من هذا النوع بالشكل "متغير.حقل=قيمة" عندما تكون الحقول من أنواع بسيطة أو قابلة للتحويل الآلي أو تم التعامل مع تحويلها يدويا فسيكون حفظ المتغيرات من هذا النوع آليا.

نأخذ مثالا على ذلك كأن نعرف هيكلا جديدا هو "Personne" له إسم و لقب و عمر و جنس. و لأجل الحقل جنس سنعرف تعدادا فيه القيمتان ذكر و أنثى بدل استعمال النوع المنطقي.

```
Enum Gender As Byte
    Male = 0
    Female = 1
End Enum
```

```
Structure Personne
    Dim Nom As String
    Dim Prenom As String
    Dim Age As Integer
    Dim Sexe As Gender
End Structure
```

```
Dim x As Personne
x.Nom = "Aloui"
x.Prenom = "Samir"
x.Age = "30"
x.Sexe = Gender.Male
```

و النتيجة كانت كالآتي:

```
<Personne>
    <Nom>Aloui</Nom>
    <Prenom>Samir</Prenom>
    <Age>30</Age>
    <Sexe>Male</Sexe>
</Personne>
```

من ملاحظة بسيطة نجد أنه صنع وسما باسم الهيكل "Personne" و صنع داخله وسوم بأسماء الخصائص و في كل وسم وضع القيمة الموافقة له.

- **الجداول** يتم تمثيلها بوسم أب هو الجدول و داخله وسوم أبناء من أجل كل خلية. يكون هناك وسم فيه النوع و القيمة كمثال على ذلك قمت بتعريف جدول من نوع مختلط و هيأته بالقيم التالية:

```
Dim x() As Object = New Object() {10, 20.2F, True, "Hello"}
```

الجدول x مهيأ بـ: قيمة صحيحة، قيمة حقيقية، قيمة منطقية و قيمة نصية بعد تخزينه في ملف xml باستعمال الـ Serialization المتوفرة في الدوت نت. حصلت على النتيجة التالية:

```
<ArrayOfAnyType>
    <anyType>10</anyType>
    <anyType>20.2</anyType>
    <anyType>true</anyType>
    <anyType>Hello</anyType>
</ArrayOfAnyType>
```

هذه الأنواع يكون التحويل فيها آليا و لا داعي لتدخل المبرمج و هناك أنواع أخرى كأن يكون كائنا مولدا من كلاس كل خصائصها من أنواع بسيطة أو من أنواع قابلة للتحويل الآلي أو لأنواع تم معالجة تحويلها يدويا

س:ماذا عن الأنواع التي تستدعي تدخل المبرمج؟

ج: هناك أنواع تستدعي تدخل المبرمج لحفظها مثل القيم من نوع Color و Font و أخرى أكثر تعقيدا مثل Image أو الملفات مثل الملفات الصوتية أو التنفيذية.

أما النوع Color و النوع Font فهما يشبهان أي كلاس أو هيكل عادي غير أن السبب لعدم تخزين قيمهما هو وجود خاصية في xml و هي أنه لا يحفظ إلا قيم الخصائص التي تقبل القراءة و الكتابة. أما في الدوت نت فإن خصائص Color و Font هي للقراءة فقط. و للاحتيال على ذلك يجب إنشاء نوع جديد مشابه للنوع الذي نريد تخزين قيمه Font أو Color و ليكن اسمه xmlColor أو xmlFont مثلا و نجعل خصائصه للقراءة و الكتابة معا. و عندما نريد الحفظ ننقل القيم من ذلك النوع للنوع الجديد و نحفظ و سيتم تلقائيا و عند القراءة ستتم تلقائيا للنوع الجديد ثم نقوم بإنشاء كائن جديد من النوع القديم و نمرر له القيم التي قرأناها كباراميترات.

كما أن هناك حالة خاصة للألوان و هي كونها يمكن تمثيلها برقم صحيح يدعى rgb أو .Argb ففي هذه الحالة لا داعي لتصميم نوع جديد و لكن فقط نعرف متغيرا جديدا و نستعمل التحويل بين القيمة Color و القيمة الرقمية في القراءة و الكتابة.

و أما للأنواع الأكثر تعقيدا كالصور و الملفات الثنائية . فالحل الوحيد هو تحويل المحتوى إلى جدول بايتات باستعمال ما يسمى بالـ Stream حيث يمكننا فتح الملف أو الصورة بواسطته بشكل جدول بايتات. إذن نستغل هذه الخاصية فنأخذ البايتات و التي قيمها محصورة بين 0 و 255 و نضعها في جدول من نوع Byte و هنا يكون حفظ الجدول آليا. و في العملية العكسية نقرأ القيم في جدول ثم نملأ Stream بقيم الجدول و نعيد بعدها توليد الملف أو الصورة انطلاقا مما لدينا

هناك نوع أخير و هو المصفوفات و التي تختلف عن الجداول الخطية و عن جداول الجداول. و السبب في ذلك أن علاقة الارتباط في xml لها اتجاه واحد فقط. لأشرح أكثر. فحين نقول "لكل شخص اسم" معناه أن الاسم متعلق بالشخص و بالتالي سيكون هناك لكل شخص اسم أي أن الوسم Name سيكون ابنا للوسم Person و لا نجد أبدا علاقة تقول" لكل اسم شخص" و لن يكون أبدا الوسم Person ابنا للوسم .Name أما في المصفوفات فهناك مثلا X و Y أو Rows و Cols و كل منهما متعلق بالآخر فنقول أن لكل سطر مجموعة أعمدة و نقول أن لكل عمود مجموعة أسطر و بالتالي علاقة الأبوة و البنوة في اتجاه واحد غير محققة و بالتالي لا يمكن تمثيلها بـxml.

التدخل اليدوي هنا ممكن بحالتين. إما تحويل االمصفوفة من الشكل [x,y,z,...etc] ألى الشكل [x][y][z]....etc] اي من مصفوفة إلى جدول جداول حيث كل جدول عناصره هي جداول و كل عنصر من تلك العناصر قد يكون أيضا جداول و النهاية تكون قيم قابلة للحفظ الآلي. بهذه الطريقة يكون الحفظ آليا.

أما لو أردنا التعامل مع الأمر يدويا فيمكن دلك بشكل بسيط و هي أن نضع كل قيم المصفوفة في جدول واحد ثم نضيف باراميترات للوسم نحدد فيها عدد الأسطر، عدد الأعمدة، و طريقة الحفظ سطرا بسطر أم عمودا بعمود.



س: قلت بأن القيم قد تخزن في بارامترات الوسوم أو كقيم للوسوم. هل هو اختيار أم هناك قاعدة تحدد مكان الحفظ؟

ج: هناك بعض القواعد البسيطة في اختيار مكان حفظ قيمة معينة. فقيم الوسوم هي القيم المهمة و الأساسية أما قيم البارميترات فهي قيم "نحتاجها في الكواليس" فهي تنظيمية أو توضيحية لأمر ما بخصوص القيم .



س: هل تعطينا مثالاتوضيحيا؟

ج: لنفرض أنه لدينا مصفوفة مكونة من 5 اعمدة و 4 اسطر مهيئة بقيم عددية. الآن أريد تخزين هذه المصفوفة في ملف xml لو قمنا بكتابة 20 قيمة مع بعض فهذا أمر عادي. لكن لو أرنا قراءة 20 قيمة لمصفوفة من ملف xml قدمه لناشخص ما فكيف سنعرف هل هي قيم جدول 1×20 أو 2×10 أو 4×5 أو 5×4 أو 10×2 أو 20×1 ؟. كل هذه القيم تمثل أبعاد مصفوفات و لكن واحدة فقط هي الصحيحة و لهذا رغم أن قيمة عدد الاسطر و الأعمدة ليست "مهمة" و لكنها صارت الآن على درجة من الأهمية بحيث لا يمكننا تعريف مصفوفة إلا منهما أو على الأقل بواسطة احداهما. حيث يمكننا تقسيم العدد الكلي على إحداهما لنحصل على القيمة الثانية. و منه فقيمة Cols و Rows لن تكون قيما للوسم مثل قيم خلايا المصفوفة بل ستكون بارامترات تستعمل في تعريف أبعاد المصفوفة.

بقي مشكل آخر بعد أن عرفنا ابعاد المصفوفة و هو هل تم التخزين سطرا بسطر أم عمودا بعمود؟ لأن القراءة بشكل يخالف طريقة الكتابة سيؤدي بنا للحصول على مصفوفة لا تشبه أبدا المصفوفة التي قمنا بتخزينها و لهذا يلزمنا باراميتر ثالث يخبرنا عن طريقة القراءة و الكتابة.

يمكن اخذ مثال آخر من HTML مثلا فنحن نحدد عرض الجدول و ارتفاعه و حجم إطاره و لون خلفيته و لكن الأسطر هي المهمة و بالتالي فالقيم السابقة هي (ديكور) تحفظ كباراميترات في وسم الجدول Table.



س: تطرقنا من قبل لطريقة حفظ قيم بسيطة باستعمال xmlText. فكيف نحفظ قيم مثل الجداول؟

ج: حفظ الجداول و القيم ذات الأنواع المركبة يكون على مراحل. فأول مرحلة يجب إضافة وسم سيضم كل القيم و الخصائص و بعدها نضيف وسوم بسيطة بها القيم البسيطة أو المركبة التي توجد في القيمة المركبة

س: هل يمكن توضيح ذلك اكثر؟

ج: سأوضح أولا بالنسبة لجدول من N خلية (بها قيم لأعداد صحيحة)

1. أولا أضيف وسما و أسميه مثلا "maTable" بالدالة CreateElement و اربطه.
2. باستعمال حلقة For من اول خلية لآخر لية في الجدول سأفعل ما يلي
   أ. أنشئ وسما و اسميه مثلا "Cellule" بالدالة CreateElement و أربطه مع الوسمmaTable
   ب. انشئ قيمة xmlText و اعطيها قيمة الخلية الحالية ثم اربطها بالوسم Cellule الذي أنشأته آخرا.

بهذه الطريقة يمكنني تخزين جدول بايتات فأخزن ملف أو صورة أو أي شيء يمكن تمثيله في جدول

س: و كيف لو أرت تخزين مصفوفة ذات بعدين يدويا؟

ج: لو أردنا تخزين مصفوفة يدويا فهناك طرق عديدة حسب الكيفية التي نريد استعمال الحفظ بها. و سأتناول 3 حالات كمثال للتوضيح. و لتكن المصفوفة التالية هي مثالنا

0	13	4	12	10
10	9	3	19	7
3	81	6	5	15

- اول كيفية هي حفظ المصفوفة بشكل سطر واحد. في هذه الحالة يجب علينا:
  - انشاء وسم جديد نسميه مثلا "maMatrice"
  - نضيف باراميترات بها عدد الاسطر و عدد الأعمدة و اتجاه القراءة و الكتابة
  - ثم نقوم بتخزين كل القيم في "نص" و نفصل بينها بالنقطة-فاصلة مثلا
  - نضع النص كقيمة للوسم.

```xml
<maMatrice rows="3" cols="5" direction="byRow">
    0;13;4;12;10;10;9;3;19;7;3;81;6;5;15
</maMatrice>
```

- ثاني كيفية و هي حفظ كل عنصر في وسم خاص به و هنا يجب علينا:
  - إنشاء وسم جديد نسميه مثلا "maMatrice"
  - نضيف باراميترات بها عدد الأسطر و عدد الأعمدة و اتجاه القراءة و الكتابة
  - باستعمال حلقة تكرارية من أول خلية لآخر خلية (حلقتين for واحدة على الأسطر و الأخرى على الأعمدة حسب ما حددناه في براميتر الاتجاه) نقوم بما يلي:
    - ننشئ وسم جديد و ليكن اسمه "Cellule" و نربطه بالوسم maMatrice
    - ننشئ قيمة بها قيمة الخلية الحالية و نربطها بالوسم Cellue الذي أنشأناه آخرا.

```xml
<maMatrice rows="3" cols="5" direction="byRow">
    <Cellule>0</Cellule>
    <Cellule>13</Cellule>
    <Cellule>4</Cellule>
    <Cellule>12</Cellule>
    <Cellule>10</Cellule>
    <Cellule>10</Cellule>
    <Cellule>9</Cellule>
    <Cellule>3</Cellule>
    <Cellule>19</Cellule>
    <Cellule>7</Cellule>
    <Cellule>3</Cellule>
    <Cellule>81</Cellule>
    <Cellule>6</Cellule>
    <Cellule>5</Cellule>
    <Cellule>15</Cellule>
</maMatrice>
```

- ثالث كيفية و هي الحفظ سطرا بسطر أو عمودا بعمود. و سأختار سطرا بسطر للمثال فسنقوم بما يلي:
  - إنشاء وسم جديد نسميه مثلا "maMatrice"
  - باستعمال حلقة تكرارية على الاسطر نقوم بما يلي:
    - ننشئ وسم جديد نسميه مثلا "Ligne" و نربطه بالوسم maMatrice
    - (يمكننا الآن استعمال طريقة دمج كل قيم السطر في نص أو يمكننا حفظ كل قيمة منفصلة و هاته سأستعملها)
    - باستعمال حلقة تكرارية على كل قيمة في السطر نقوم بما يلي:
      - ننشئ وسما جديدا نسميه Cellule و نربطه بالوسم Ligne الذي أنشأناه آخرا.
      - ننشئ قيمة و نضع به قيمة الخلية الحالية و نربطها بالوسم Cellule

```xml
<maMatrice>
    <Ligne>
        <Cellule>0</Cellule>
        <Cellule>13</Cellule>
        <Cellule>4</Cellule>
        <Cellule>12</Cellule>
        <Cellule>10</Cellule>
    </Ligne>
    <Ligne>
        <Cellule>10</Cellule>
        <Cellule>9</Cellule>
        <Cellule>3</Cellule>
        <Cellule>19</Cellule>
        <Cellule>7</Cellule>
    </Ligne>
    <Ligne>
        <Cellule>3</Cellule>
        <Cellule>81</Cellule>
        <Cellule>6</Cellule>
        <Cellule>5</Cellule>
        <Cellule>15</Cellule>
    </Ligne>
</maMatrice>
```

يمكن استعمال الحفظ عمودا بعمود أيضا بنفس الطريقة. تقنيا فلو كنا سنقوم بالقراءة اليدوية فالأحسن اختيار الكيفية الثانية لأن هذه الكيفية تسمح لي بمعرفة عدد الأسطر و الأعمدة من البداية فأستطيع مباشرة تعريف أبعاد المصفوفة عند التصريح بها كما أنني أستطيع استعمال حلقتي For لقراءة القيم طالما أنني أعرف قيم التوقف للحلقة.

س: في ماذا يستعمل النوع CDATA ؟

ج: قيم هذا النوع هي قيم نصية مثل قيم xmlText و لكن السبب في إضافتها هي كونها تستطيع حفظ النص تماما كما هو دون ترجمته .



س: هل يمكن التوضيح اكثر؟

ج: كما نعلم فالملف xml هو ملف نصي و لكي يتعرف عليه الـ Parser فإنه يستخرج الكلمات المحصورة بين < و > ليعتبرها Elements و يستخرج أيضا البارامترات و يتعرف على التعاليق بـ <!-- و غيرها من الرموز التي تتيح له فصل الكلمات المفتاحية من القيم.

لكن ماذا لو أردنا تخزين جملة تحتوي على وسوم كأن نريد حفظ ملف xml كعنصر أو نريد حفظ صفحة html كعنصر. أو نريد حفظ أي شيء يكون فيه وسوم أو بعض الرموز التي قد تكون رموز مفتاحية لـ xml. فلو حفظناها في xmlText فسيتم ترجمة تلك الكلمات لاحقا و بدل أن تكون نصا ستعتبر وسوم جديدة و سيقع تأثيرها على الملف ككل و قد يصبح الملف خاطئا في حال وجود محتوى ترجمه كوسم مفتوح بينما لا يوجد وسم يغلقه مثلا.

لهذا السبب بالذات جاء النوع CDATA و وسمه يدعى xmlCDataSession و يتم توليده بالعبارة CreateCDataSession حيث يمكننا تخزين أي شيء به حتى لو كان نصا لملف xml آخر. و عند القراءة فإن الـ Parser لن يقوم بترجمة أي شيء داخل هذا الوسم و سيعيد المحتوى على أساس أنه نص.



إلى هنا نأتي لنهاية أول موضوع من السلسلة . في انتظار ملاحظاتكم، مشاركاتكم و انتقاداتكم استودعكم الله الذي لا تضيع ودائعه و السلام عليكم و رحمة الله و بركاته